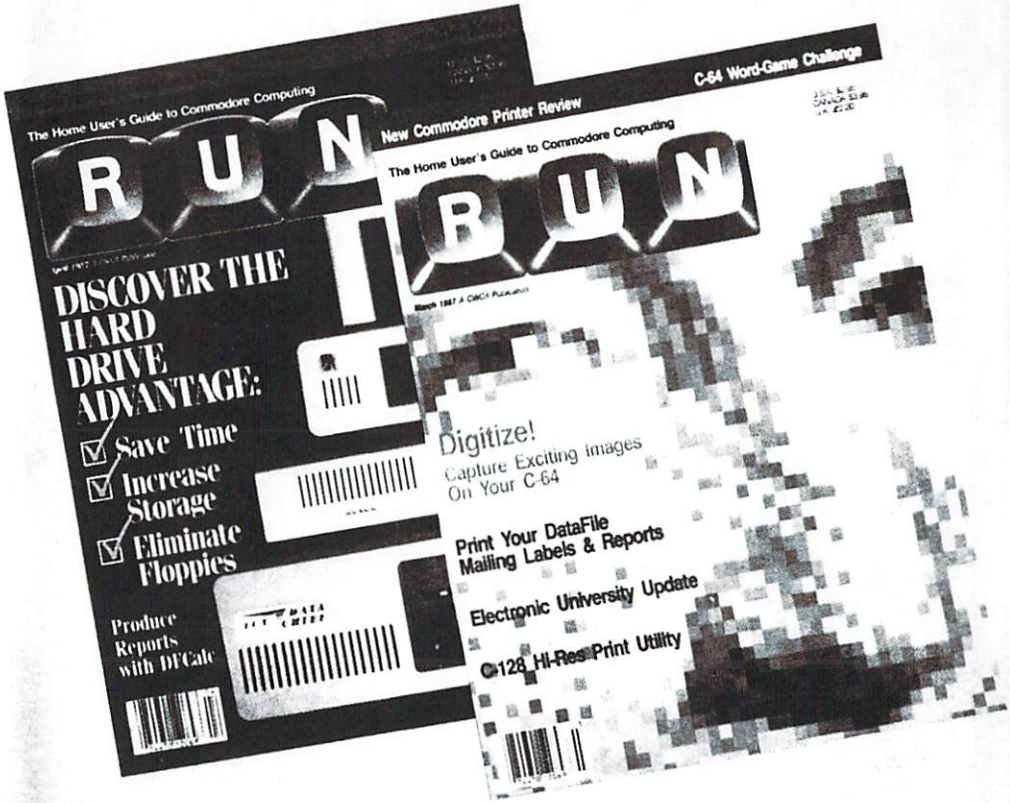
March/April 1987 Edition

RUN

RUN Programs on Disk

For the C-64 and C-128*



Plus: Bonus Program!

*128 mode programs included

Introduction

March-April '87 ReRUN

The wait is over. Here's the March-April 1987 edition of ReRUN, containing all the programs from the March and April issues of *RUN* magazine, plus Datafile, as a special bonus program. Let's take a look at the programs from the March issue.

We'll begin with DFPrint, the ultility that allows you to print the records you create with Datafile 3.6, one of *RUN*'s most popular programs to date. DFPrint creates mailing lists, prints your records, either formatted or unformatted, and allows you to save the print formats.

Pick your Ps and Qs is both an educational and an entertainment program. It expands your vocabulary by letting you choose letters from an alphabet, located at the bottom of the screen, to spell a word defined at the top.

In the area of easy applications, there's The Commodore in the Kitchen. A clever recipe converter, it allows your Commodore to convert army mess hall-volume recipes down to a size fit for a family of four and vice versa.

Commodore 128 owners will appreciate March's Mega-Magic program—C-128 Screen Dump. With this print utility, you'll be able to print those 40-column hi-res screens that your C-128 is capable of producing.

April programs include DFCalc, the calculation accessory program for Datafile 3.6. Use it to create spreadsheet-like reports using your Datafile files. The Menu Machine, our April Easy Applications program, sheds light on a new approach to designing menu-driven programs. Finally, April's Mega-Magic is C-64 Edit Enhancer, a useful utility for toggling your C-64 in and out of Quote mode.

For a bonus program, there's Datafile 3.6, published in February of this year. Included with Datafile 3.6 are two short sub-programs, one for creating a sorter and the other for making DOS 5.1 compatible with Datafile.

Let me know what you think about ReRUN. Keep me informed, and I'll continue to do my best to meet your needs.

Tim Walsh
Technical Editor
RUN *magazine*

How To Load

Loading from Menu

To get started, C-64 users should type LOAD "MENU 64",8 and press the return key. When you get the Ready prompt, the menu is loaded and you should type RUN to see a list of the programs on your disk.

Loading from Keyboard

If you do not wish to use the menu program, follow these instructions.

C-64:

To load a C-64 program written in Basic, type:

LOAD "DISK FILENAME",8

and then press the return key. The drive will whir while the screen prints LOADING and then READY, with a flashing cursor beneath. Type RUN and press the return key. The program will then start running.

C-128:

All C-64 programs can be run on the C-128 as long as your computer is in C-64 mode.

All C-128 programs are clearly labeled on the directory page. Your C-128 *must* be in C-128 mode to run these programs.

To load a C-128-mode program, press the F2 key, type the disk filename and then press the return key. When the program has loaded, type RUN.

Making Copies of ReRUN Disks

Some of the programs on your ReRUN disk have routines that require you to have a separate disk onto which the program writes or saves subfiles. In order for you to use these programs, you will first have to make a copy of the original program onto another disk that has enough free space on it to hold these newly written subfiles.

If the program is written in Basic, it is simple to make a copy of the program. Just load the program into your computer following the procedures outlined above, and then save the program back onto a separate disk that has plenty of free space for extra files.

Directory

Page		Article	Disk Filename	File Type
			MENU 64	BASIC
1	£	Datafile 3.6	INSTALL DOS5.1	BASIC
			SORT GENERATOR	BASIC
			DATAFILE 3.6	BASIC
6		DFPrint	DFPRINT	BASIC
14		DFCalc	DFCALC	BASIC
29		Pick Your Ps and Qs	PICK A LETTER	BASIC
31	*	C-128 Screen Dump	C128 SCREEN DUMP	BASIC
32		Mega-Magic, March	FILE LOCK	BASIC
33		The Commodore in the Kitchen	RECIPE	BASIC
37		The Menu Machine	MENU PRG. 1	BASIC
			MENU PRG. 2	BASIC
			MENU PRG. 3	BASIC
40		Mega-Magic, April	64 EDIT ENHANCER	BASIC

* C-128 program

£ Bonus program

NOTE: Read articles before running these programs!

ReRUN Staff

Technical Editor: *Tim Walsh*
Managing Editor/Production: *Swain Pratt*
Copy Editor: *Peg LePage*
Proofreader: *Harold R. Bjornsen*
Design and Layout: *Karla M. Whitney*
Typesetting: *Doreen Means, Beth Krommes, Ken Sutcliffe*
Special Products Director: *Jeff DeTray*
Special Products Manager: *Vivian Mattila*
Special Products Assistants: *Debbie Bourgault, Robyn Johnson*
Direct Marketing Coordinator: *Debbie Walsh*

Datafile 3.6

By Mike Konshak

RUN It Right

C-64; disk drive; printer

Datafile 3.6 is the newest version of the Datafile database-management system for the C-64—a memory-based system that uses sequential files.

Datafile, the main program of a three-part package, will not alone allow printing of records, but only writing to disk. DFPrint and DFCalc, the two accessory programs that print out Datafile 3.6 files, are on this disk and elsewhere in this booklet.

With Datafile, you can create your own database, choosing the number and length of fields, as well as their titles. After you've created a record file and entered your data, the program will search, sort, delete and modify the records.

As you work with Datafile, keep in mind that it will accept only non-shifted characters when you're inputting data, and that commas, colons, semicolons and quotation marks are not allowed.

If you're adding a large number of records to a file at one sitting, be sure to save the file often, just in case the power goes out unexpectedly.

DATAFILE INSTRUCTIONS

Datafile 3.6 uses an ML fast sort routine and the DOS 5.1 wedge. First, save DATAFILE 3.6, SORT GENERATOR and INSTALL DOS 5.1 to a newly formatted disk. Next, run SORT GENERATOR. When run, this creates and saves the ML sort file to disk. Lastly, run INSTALL DOS5.1. This lets you copy DOS5.1 from your Test Demo disk onto your Datafile disk.

After doing this, you are ready to use Datafile. Just type LOAD "DATAFILE",8 and RUN. The two utilities will load and run, and the main menu will appear as follows:

```
CREATE NEW FILE
QUIT PROGRAM
ADD RECORD TO CURRENT FILE
MODIFY RECORD IN CURRENT FILE
DELETE RECORD IN CURRENT FILE
VIEW OR EDIT FILE
```

SORT RECORDS BY FIELD
PRINT RECORDS USING DFPRINT/
DFCALC
READ (LOAD) OLD FILE FROM DISK
WRITE (SAVE) CURRENT FILE
TO DISK
@ DISK DRIVE COMMANDS
$ 4 DIRECTORY

You choose a menu option by pressing the key for the first letter of the option. If your program ever crashes or locks up because of a disk drive or printer error, just type GO 68 <return> to get back to the main menu without losing any record data.

CREATING A NEW RECORD FILE

Now you need to create a datafile, so enter C for the Create option. When you create a record file, you're defining the structure to which all its records must conform, so evaluate carefully the needs of your application. It's rather difficult to change your mind later, although it's possible by using Datafile utility programs. (See *RUN*, November 1985, for the DFRestructure utility.) The rules for creating record file structures are as follows:

1. It's advisable to have no more than 15 fields.
2. As indicated above, field titles cannot contain quotation marks, commas, colons or semicolons.
3. Field length, including the field title, cannot exceed 80 characters.

Now let's create a sample record file for keeping track of club members. The file MEMBERS will have the following structure:

FIELD	TITLE	LENGTH
1	LAST NAME	15
2	FIRST NAME	15
3	STREET	30
4	CITY ST	22
5	ZIP	7
6	PHONE	12
7	DATE JOINED	8

Enter this information and then watch for the display that tells you how many records the structure can hold. If the structure is a good size and otherwise satisfactory, press A to accept it. We'll assume this sample structure is all right. You now have a current file in memory.

MODIFYING RECORDS

Now, type about ten records into your file for some data to experiment with. When you press M, for modify, on the main menu, Datafile will ask which records you want to change. If you want to change just one, and you know its number, type the number and press return. If you don't know the number or you want to change a number of records, press A to view all the records in the file, one at a time.

The current data in each field in each record will be displayed in turn. If you want to leave the field as is, press return. If you want to erase its data, press the > key. (The > will stay within that record until the file has been saved and reloaded.) You can also copy a field's data by pressing the equals key. At the end of each record, press N to advance to the next record or E to exit.

Unless you know the record number or have changes to make on every record, it's more convenient to use the alternative method of modifying records available through the View or Edit option on the main menu.

DELETING RECORDS

When you press D for deleting records, you once again have to designate a record or press A to go through the whole file. If you can't remember the record number, go to the View or Edit option on the main menu and delete the record from there.

Before Datafile will delete a record, it displays the record's entire contents on the screen. If you're sure you want to delete it, press shift/D.

As the deletion occurs, the count of records in the file decreases, and all the records after the deleted one are renumbered accordingly. To put all your records back into order, you have to use the Sort option in the main menu. Remember to save your revised file to disk.

VIEW OR EDIT FILE

This option offers the most flexibility for viewing, scanning and editing the current file. As each record is displayed, you'll be given the following eight choices:

NEXT	LAST	JUMP	FIND
MODIFY	DELETE	PRINT	EXIT

Next makes the screen step to the next record, Last steps it backward to the previous record, and Jump takes it directly to a particular record number, instead of stepping one by one. Print sends the record currently on the screen to your printer.

Find lets you locate records having common data within a certain field. Then you can modify or delete each record. When you're using Find, the screen displays a list of the field names in your current datafile and asks you to enter the number of the one you wish to search. The field name is then displayed, and you must enter the common item. Type in the string of text you're looking for and press return.

For example, if you choose a first-name field, you might enter the string JIM. The computer

would search out all the records that begin with JIM in the first-name field. Not only would JIM come up, but also JIMMY, because it begins with JIM.

SORTING

When you pick the Sort option at the main menu, the screen displays the names and numbers of the fields in the file that's in memory. You can sort the file by up to five fields, all in ascending order, and the sort will take less than 10 seconds.

Datafile stores all data as strings, not as actual numbers. For this reason, the value of each field, when compared for sorting, is determined by the position of each character. Therefore, be sure to be consistent with the format when you're entering field data.

WRITING FILES TO DISK

To save (write) a file to disk, choose W at the main menu. Datafile will ask for the name of the file, and save the file after you respond. The name may be up to 12 characters long.

When you save a record file, Datafile automatically attaches the four-character prefix DF] <space> to the filename. For instance, the name of your sample file will become DF] MEMBERS. This prefix will show up when you list the directory of the files, but you usually won't need to use the prefix yourself.

Any time a record file is written onto a disk where a file with the same name resides, Datafile makes the earlier version a backup and assigns it the suffix .BAK. Therefore, when your sample file is saved the second time, the first version will be retained on the disk with the name DF] MEMBERS.BAK.

You can load the earlier version from the Read Old File option on the main menu. To do so, enter only your filename with the suffix—MEMBERS.BAK here. Don't include the prefix that shows in the directory.

Datafile keeps only one generation of backups, so the third time you save MEMBERS, the first version will disappear. If, for some reason, you want to keep more than one generation of backups, you must give the older ones a different filename.

READING FILES FROM DISK

You'll usually pick the Read option from the main menu at the start of a Datafile session. It loads a file you've saved previously. After you've entered R, the program displays all the available files and asks which one you want to load. Type in its name and press return. The file will load, and Datafile will return to

the main menu. The program will also return to the main menu if you press the return key without typing a filename.

Remember, don't type in the four-character prefix when you enter a filename to be read; just type the name to the right of the bracket and space.

DISK COMMANDS

Datafile has five disk commands. You access the disk command menu, which contains these options plus another that returns you to the main menu, by pressing the @ key at the main menu. The disk-command options are as follows:

FORMAT

This feature allows you to format a blank disk to use for saving files. Insert the disk into the drive, then enter a disk name (up to 16 characters long), a comma, and a 2-character disk ID (any combination of numbers and letters)—for example DATAFILE FILES,D2. Follow this sequence with a return. The drive will whir for about 3½ minutes while it's formatting the disk, then return you to the main menu.

Make sure the disk you place in the drive for formatting is really the one you want to use, because this process will erase the entire disk!

DISK DIRECTORY

To list the directory of the disk currently in the drive, press the 4 key. After you've finished viewing the directory, press any key to return to the disk menu.

SCRATCH A FILE

To scratch any sequential file on the disk, enter the filename, including the DF]<space> prefix, at the prompt and press return. For instance, to scratch your sample file, you'd type DF] MEMBERS. Be sure to type in the name *exactly* as it appears in the directory, so you don't scratch the wrong file by mistake.

RENAME A FILE

To rename a sequential file, enter the old name exactly as shown in the directory, then the new name when the prompt appears. Be sure to include the special prefixed characters; otherwise, Datafile won't recognize the newly named file, and you won't be able to load it from the main menu.

VALIDATE A DISK

This option removes any corrupted files (splat files, with an * beside them in the directory) from your disk.■

DFPrint

By Mike Konshak

RUN It Right

C-64; printer

DFPrint is the first of the two major accessory programs you need to print out Datafile records. Load DFPrint and run it. The configuration menu described at the end of this article appears first. Choose among options 1–4 before continuing.

DFPrint will then present these options:

```
DATAFILE FAST PRINT PROGRAM

OPEN RECORD FILE ON DISK
$ DISK DIRECTORY 4
QUIT PROGRAM

TRANSFER TO:
  D DATAFILE
  C DFCALC

PRESS THE APPROPRIATE KEY
```

DFPrint assumes that you'll be printing a record file that was created, edited and written (saved) to disk with Datafile and that the record file is located on the disk in the disk drive. It prints the records in the order in which they were last sorted and saved by Datafile. DFPrint can only read record files, and so cannot make any changes to the records.

Pressing $ or 4 lets you see the disk directory, and Q quits the program altogether. Pressing O invokes a prompt for you to enter the name of the Datafile to be read. Insert the disk containing your records, type in the name of your record file, MEMBERS, and press the return key. The program will read the structure of the file to get information it will use later. After the light on the drive goes out, you'll see the following menu:

```
RECORD PRINTOUT MENU

PRINT OPTION: NO FORMATS
      PRESENT
  UNFORMATTED LIST
  REPORTS FORMAT Formats: RP]*
  MAILING LABELS        : ML]*

CONFIGURE:
  LABELS: 5 ROWS, 34 CHRS,
    1 ACROSS
  PRINTER: OPTION 4, DEVICE# 3
  EXIT TO RESTART

PRESS THE APPROPRIATE KEY
```

This screen lets you choose the type of printout that best suits your needs and reconfigure the printer or change label sizes and type. You can also display the disk directory from this menu.

PRINTING UNFORMATTED LISTS

This is by far the easiest printout option, because you don't have to create a format. Records and fields will be printed in much the same way that you might fill out 3-by-5 cards. An unformatted report resembles the following:

```
[ RECORD # 1 ]----------
NAME . . . . . .MIKE
AGE . . . . . .39
BIRTHDAY . . . . . .05/28

[ RECORD # 2 ]----------
NAME . . . . . .BECKY
AGE . . . . . .27
BIRTHDAY . . . . . .06/27

[ RECORD # 3 ]----------
NAME . . . . . .SARAH
AGE . . . . . .11
BIRTHDAY . . . . . .09/10
```

The name of each field is printed along with the field data and record number. The program will automatically skip over the perforations by calculating how many complete records will fit onto one page. The dashed line is printed across the entire page as a guide for cutting the paper in case you wish to attach the record data to a card. This type of printout uses a considerable amount of paper, so it's best to define a report format for printing large files.

If your field lengths exceed 60 characters, you'll have to put your printer into a compressed mode. This prevents the record data from wrapping around to the next line, which would throw off the pagination count.

SELECTING RECORDS

After you've chosen the Unformatted List option or entered a print format into the computer from the disk, the disk drive light will turn on as the drive positions itself at the start of your record file. Then, using the following screen, select which records to print out:

```
PRINT OPTIONS MENU

THERE ARE 22 RECORDS IN
  MEMBERS

ALL RECORDS IN FILE
  FIND RECORDS WITH COMMON
    FIELDS
EXIT BACK TO START

PRESS THE APPROPRIATE KEY
```

Pressing A prints the entire file using the current format. You'll first be asked the starting record number. Record 1 is the default starting position.

Pressing F invokes a routine that searches for records according to your entered data. After choosing the field to be searched, enter the data to which the records will be compared. If the beginning of the record field matches your data, the record will be printed out.

Pressing the return key during printing stops the process after a record is completed. You can continue printing or exit the routine, as prompted.

PRINT-FORMAT OPTIONS MENU

Whenever you choose M for mailing labels or R for reports at the record-printout menu, you'll be confronted with the following screen:

```
PRINT OPTIONS MENU

CURRENT RECORD FILE: MEMBERS
CURRENT FORMAT FILE:
  LOAD OLD FORMAT
  CREATE/CHANGE FORMAT
  SAVE CURRENT FORMAT
  PRINT RECORDS
  DIRECTORY $
  EXIT

PRESS THE APPROPRIATE KEY
```

If you're accessing this routine for the first time and no print format is present, then two of the options, Print and Save, will not be displayed. If you have a format on the disk, then you can load it by first pressing L and then entering the format filename as prompted.

If you've never created a print format for the type of printout you want, or if you desire to change one that's already loaded, you must press C to enter the respective format-definition routines. Creating formats will be discussed in subsequent sections.

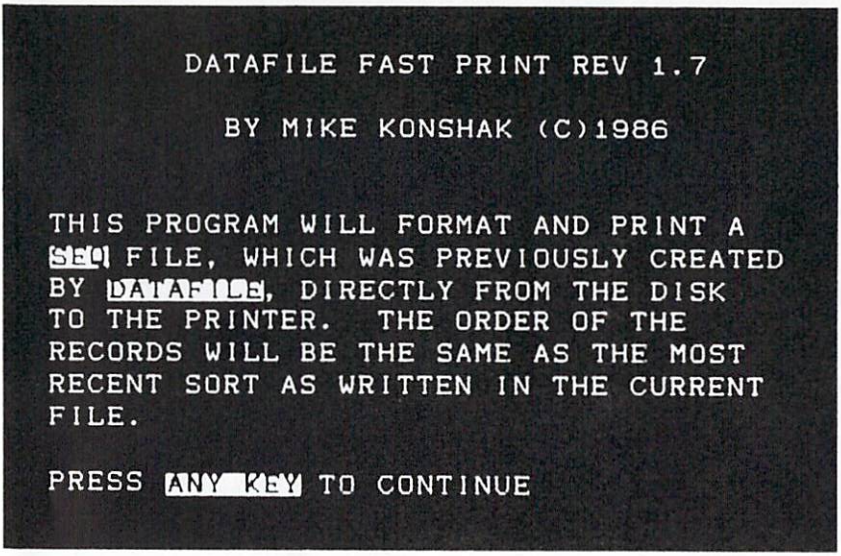
Once a print format is present in memory, pressing P will advance you to the Print Options menu, which selects records for printing. This menu was discussed above.

MAILING-LABEL FORMATS

Before records from your file can be printed out on labels, you must define a label format specifically for your file. Formats tell the computer which of your record fields to print in each row of the label. Besides the actual data that's to be printed, you must define the size of your label.

The size of a label is specified by the number of rows and the number of characters in each row. Datafile defaults to a standard label size, which measures $^{15}/_{16}$ of an inch wide and 3 inches long. This popular label size can be printed with 5 rows and 34 characters.

There are various sizes and types of labels available. By



types, I mean the number of labels across a page, sometimes referred to as one up, two up, and so forth. Datafile will print up to 30 rows and 136 characters per row on labels up to four across a page (four up). You should be able to design a format around any size or type of label. I prefer one-up labels, but some printers don't have adjustable tractors, in which case you must print the labels two (or more) up.

After defining the number of rows, you'll be asked which fields you want printed in each row. Up to three record fields may be printed in each row and in any order. You'll be required to enter at least one record-field number for the first field in each row that will print data. If you want to skip a row (print a blank line), you must enter a zero in all three fields for that row.

When asked for fields 1–3 in each row, enter the record-field number behind the prompt. A zero will be preprinted for you the first time. If you only want one record-field in a row, enter the desired record-field number for field 1 and zeros for fields 2 and 3.

For example, let's use the address record file called MEMBERS for creating a label format. MEMBERS is a good candidate, since it needs to combine multi-

ple fields in several rows. MEMBERS has the following structure:

Field	Title	Length
1	LAST NAME	15
2	FIRST NAME	15
3	STREET ADD	30
4	CITY ST	22
5	ZIP	7
6	PHONE	12
7	DATE JOINED	8

Knowing the record file structure, we'll create a label with five rows. The date the member joined will be in row 1, so he or she will know when it's time to pay dues. We'll skip row 2, the first name and last name will be in row 3, the street address in row 4, and the city, state and zip code in row 5. The rows would look like the following:

Row	Field1	Field2	Field3	Will print
1	7	0	0	DATE JOINED
2	0	0	0	
3	2	1	0	FIRST & LAST NAME
4	3	0	0	STREET ADDRESS
5	4	5	0	CITY ST & ZIP

As you can see, it's not important which position the record fields are in, because you can rearrange them when you create the label format. Again, let me remind you that the default label size is 5 rows, 34 characters long and one up. If you're using a different size or type of label, you must alter the label configuration by pressing L at the print-options menu.

Label formats are handy if you wish to review all your records on the screen. Use printer option 4 and one of the multiple-up options, and change the number of characters to get as many records as possible across the screen.

When you finish defining your label format, you'll have an opportunity to review and change the format. If you press N, you'll be asked if you want to save the label format. You should do so immediately, lest you forget it later. Press Y and you'll be prompted with the current record filename. Either press the return key to accept the filename as is, or make the appropriate changes. Use no more than 12 characters for your label-format filename. A label format with the same name will be scratched and replaced with the new format file. Label format files appear on the disk in the form ML] MEMBERS.

CREATING REPORT FORMATS

Just like labels, a report format must be designed around your record file if you wish to print the

data in an organized and well-presented columnar report. Reports differ from labels in that the record-field data is printed in columns instead of rows. This report-format routine is recommended for reports containing mostly text, although the last column may be used for totaling numbers. Use the DFCalc program (see page 14) for reports that require a lot of calculations. In every report, you have the capability for, and must define, the following parameters:

Width in characters. Up to 136 characters can be printed across the page, depending on the capabilities of your printer. You might have to put your printer into a compressed print mode to print more than 80 characters across the page.

Number of title lines. You can have up to four titles printed at the top of the page. The titles will appear only on the first page.

The title for each title line. Titles can be as long as the width of the report. They may not contain commas, colons or semicolons. Examples of titles are: company name, date, report information and sources.

Number of columns. You can have up to 16 columns, at least one of which must be printed.

Header for each column. Headers identify the contents of the column and may be any name you choose. They are printed above every column at the top of every page. They cannot be wider than their respective columns, nor can they contain commas, colons or semicolons.

Position of each column. You must specify the starting location of each column—a number from 1 to the width of the report. You'll have to determine beforehand how wide you expect each column to be, based on the length of the record-field data that will be printed in the column. In doing your calculations, keep in mind that the columns will be separated by two spaces. You may have to go back and adjust the width of the report and the position of the columns after you see your first printout. A sheet of graph paper comes in handy here.

Contents of each column. Up to three record fields can be printed in each column. The fields within a column will be separated by one space, and the entire contents of the column will be printed left-justified. At least one record field must be chosen for each column and entered in the first field position. As in formatting labels, enter zeros for any unused fields. The record-field titles will be printed on the screen for easy reference. You can print only one record in each row of a report.

Total the last column? The only calculation the report format will perform is summing the contents

of the last column. The record-field data defined for the last column must be numeric and must be entered into the first field position. If nonnumeric data (like the dollar sign) occurs at the beginning of the record field, the program will produce a result of zero for that record. The total of the column will be printed at the bottom of the report. The values in the column will be printed right-justified with two decimal places, such as 125.50. To select the Totaling option, choose 1 when prompted at the end of the formatting process. Entering 0 turns off totaling.

When you're finished defining your report, you're asked if you want to go back and review or change the format parameters. If you respond with N, you're asked if you want to save the format. If you've made any changes at all to an existing format, or have created a new format, press Y at this time. Use the preprinted filename or type in a new name. Use a maximum of 12 characters in the name of the report-format file. If a file with the same name already resides on the disk, that file will be scratched, to be replaced by the new one.

Using the file MEMBERS as an example, you might create a report format using the following parameters:

```
REPORT FORMAT FILE: MEMBERS
  (appears on the disk as
  RP] MEMBERS)
NUMBER OF CHARACTERS (wide): 80
NUMBER OF TITLE LINES: 2
TITLE 1: CLUB MEMBER ÅDDRESS
  LIST
TITLE 2: MAY 31 1985
NUMBER OF COLUMNS: 5
CLM 1: POSITION = 1 HEADER =
  FIRST/LAST NAME FIELDS = 2 1 0
CLM 2: POSITION = 19 HEADER =
  STREET ADDRESS
FIELDS = 3 0 0
CLM 3: POSITION = 41 HEADER =
  CITY AND STATE FIELDS = 4 5 0
CLM 4: POSITION = 60 HEADER =
  PHONE NUMBER FIELDS = 6 0 0
CLM 5: POSITION = 74 HEADER =
  JOINED FIELDS = 7 0 0
TOTAL LAST CLM: 0
```

Enter the above values after the respective prompts. You'll notice that after you've entered a value, it will be preprinted for you when you go back to review the format parameters.

PRINTER INTERFACE CONFIGURATION

Because there are so many printers that will attach to the C-64 computer, Datafile can be configured to your special system. Upon accessing DFPrint or DFCalc, you'll be given a choice of four print configurations. The option display will appear as follows:

```
PRINTER-INTERFACE CONFIGURATION
CURRENT OPTION IS 4
PRESS 1 CARDCO A,C = 1525
      2 PRINTERS W/GRAPHIC
        INTERFACES
      3 C = 1526, C = MPS801/802/803
      4 PRINT OUTPUT TO SCREEN
        OR SEND PRINTER
        COMMANDS
      EXIT TO CONTINUE

PRESS THE APPROPRIATE KEY
```

Options 1–4 are for choosing the configuration. The current option will change each time you press one of these numbers. After selecting one (such as 4, for printing to the screen), you can choose to send commands to your printer or exit this menu into the next part of the program.

SENDING PRINTER CODES

Pressing S from the printer-interface configuration menu will deliver you to a routine for sending ASCII codes to set up your printer for various print modes and styles. Most Commodore printers don't offer much flexibility with compressed print and various type fonts.

You'll have to check your printer manual for the codes you can send. Look for statements that print numbers within CHR$( ) commands, such as PRINT#4, CHR$(15), which makes a Commodore printer print all double-wide or enhanced characters, or PRINT#4,CHR$(27)CHR$(66) CHR$(2), which invokes a compressed print mode (12 characters per inch) on Star printers. Commands that look like PRINT#4, ESC "W"1 must be converted to their ASCII equivalents, such as PRINT#4,CHR$(27)CHR$(87) CHR$(1).

When you're sending printer command codes, enter only the numbers within the parentheses, such as 27, 66 and 2. Enter the numbers one at a time, following each number with a return. You may send one to four numbers, then terminate the sequence with an asterisk (*). Your printer will probably do a linefeed at this point, indicating it has received something.

You can test your change by pressing T at the end of the routine, and you can also choose to send another code by pressing A. E will return you to where you left off in the particular program you're in.■

DFCalc

By Mike Konshak

RUN It Right

C-64; disk drive; printer

DFCalc is the second accessory program to Datafile. You can load DFCalc from the main menu of Datafile by pressing P, for Print option, and then C in the resulting option menu; or load it in from Basic by typing LOAD "DFCALC",8 <return> and RUN <return>.

DFCalc enables you to use your Datafile record-file data to produce spreadsheet-like reports. Since the program provides for considerable flexibility in designing your calculated reports, many applications are possible.

Calculations are performed on only one record at a time within a row of the report. However, the entire contents of a column can be totaled or averaged. Various methods of justification are provided, as well as many mathematical tokens to expand the capabilities of your printed document.

The following is a comprehensive description of DFCalc's capabilities:

1) A header consisting of up to four title lines can be printed at the top of the first page of your report.

2) You can define up to 16 columns in your report. The number of columns possible will depend on the width of each column and the number of characters your printer can fit across the page.

3) Columns can contain:

a) The record number.

b) The contents of a field or data within a record.

c) The contents of another column within the current row.

d) An equation, the results of which will be printed within the column.

e) A running total of the data found in a previous column.

4) Equations will perform operations using the following operands:

a) Numerical content of a field within a record.

b) Numerical content of a previous column within the row.

c) Numerical constants or values, such as 100, 3.14159...(pi), 469.65, 2.6769E – 3, 8.965E10, and so forth.

5) Equations can perform operations employing up to three of 47 operators, such as +, –, ×, ÷, ↑, sin( ), log( ) and sqr( ). Included among these are conversion operators that will convert inches to millimeters, degrees to radians, and so on.

6) End of Column (EOC) operations can be defined as follows:

a) No operation (NOOP), which doesn't print any values at the end of a column.

b) Calculate the sum of all the data printed in the column.

c) Calculate the average of all the data printed in the column (the total divided by the number of records printed).

7) Column justification can take the following forms:

a) Left (for alphanumeric data).

b) Right (for alphanumeric data).

c) Currency (numeric data, right-justified, two decimal places).

d) Decimal (numeric data, right-justified, zero to eight places).

e) Percent (multiplies contents of column by 100).

STRUCTURE OF DATAFILE * INVENTORY *

\# RECORDS POSSIBLE IN FILE: 253
\# RECORDS IN CURRENT FILE: 5
\# FIELDS IN EACH RECORD: 6

#	TITLES OF FIELDS	LENGTH
1	ITEM	20
2	DESC	20
3	SER#	20
4	COST	9
5	YEAR	2
6	ROOM	10

Figure 1.

DFCALC INSTRUCTIONS

DFCalc will produce accurately calculated reports from your Datafile record files, assuming you design the report format correctly. For the purpose of learning the program, we'll create a fictitious record file as an example.

DECIDING ON A REPORT

Let's assume you've previously created a record file called Inventory with Datafile. It will appear on the disk directory as DF] INVENTORY. The special characters preceding the right bracket identify the sequential file as having been written by Datafile. The Inventory file will have the structure shown in Figure 1. The five

REC#	ITEM	DESC	SER#	COST	YEAR	ROOM
1	COMPUTER	COMMODORE.C64	S00011142	595.00	83	DEN
2	TV	SONY	HA778800	365.50	72	FAMILY
3	TAPE.DECK	SHARP	LG324M01	129.95	77	FAMILY
4	MICROWAVE	GENERAL.ELEC	444TT6789	435.00	84	KITCHEN
5	CAT	CRUISER	>	>	81	

Figure 2.

COLUMN:	1	2	3	4	5	6	7	8
HEADERS:	REC#	ITEM	DESCRIP	SERIAL#	COST	YEAR	AGE	REPCOST
POSITION:	1	5	16	29	43	52	56	61

Figure 3.

COLUMN	CONTENTS	EQUATION	EOC	JUSTIFY
1	#		N	R
2	F1		N	L
3	F2		N	L
4	F3		N	L
5	F4		T	C
6	F5		N	L
7	E	86–F5	A	R
8	E	((F4*.05)*C7)+F4	T	C

Figure 4.

		NUMBERS	TEXT
L	— LEFT justify	(123)	(ABC)
R	— RIGHT justify	(123)	(ABC)
C	— CURRENCY, 2 places	(123.00)	(.00)
D4	— DECIMALS, 0–8 places	(123.0000)	(.0000)
P1	— PERCENT, RES*100	(12300.0)	(.0)

Figure 5.

records in the sample file appear in Figure 2.

Now that you have a file and know what the structure is, let's determine what to do with the data when you enter DFCalc. The records in Inventory will be used to keep a file of valuables for insurance purposes. Besides the actual records in Inventory, it might be useful to determine the replacement cost of each item. This would require adjusting the original cost by the age of the item and the yearly inflation rate, thus introducing two additional data fields, each containing a formula.

REPORT LAYOUT

Figure 3 suggests the layout of the report, which is 67 printout columns wide. Regarding the fields as "columns," they are numbered 1 through 8. The field descriptions, or column headers, are below the column numbers, and the bottom line, labeled "position," gives the printout column in which each report column begins. There are two spaces between columns. If your record field data is longer than the width of a column, the data will be truncated, starting at the right.

Once you've positioned the columns, you must determine what contents and operations will be part of each column. Figure 4 shows the structure of each column. I've explained the criteria above.

CONTENTS OF COLUMNS

In the example, INVENTORY, the record's number is printed in the first column, using #. Columns 2 through 6 will contain whatever is found in the respective fields for that record number, indicated by F1 through F5.

Columns 7 and 8 are defined to have formulas, or equations, denoted by E. Column 7's equation involves subtracting the value found in column 6 (F5, the year of purchase) from the constant 86 (the current year). This results in the age of the item. Column 6 (C6) could be used in place of F5 as notation in the equation, since that column contains the field data. The formula would then be 86 – C6.

Column 8's equation uses the maximum number of operators allowed within one column, three, to derive the replacement value. The original cost (F4) is multiplied by .05 (the yearly inflation factor), and this result is multiplied in turn by the age of the item as calculated in C7. Finally, this result is added to the original cost (F4).

Column 8's formula could also have been written ((C5*.05)*C7) + C5. Since there were no calculations performed in column 5 (C5), the column and the field have the same value.

TOTALS AND AVERAGES

Still referring to Figure 4, the End of Column (EOC) operation must be defined with one of the following three options:

N—*No operation (NOOP)*—do nothing with the column.

T—*Total*—find the sum of the contents of the column.

A—*Average*—find the average of the contents of the column (average = total ÷ number of records).

Since columns 1–4 and 6 are just displaying field data, we choose no End of Column operations, using N. Column 5 contains the original cost, so we choose T to obtain the total of the original cost of all the items. Column 7 calculates the age of all the items, so it might be interesting to see the *average* age of all our goods by entering A for that column. We choose T for column 8 simply because we'd like to know how much to ask the insurance company for (or at least to realize how much new things cost nowadays).

JUSTIFICATION OF CONTENTS

You must choose how the field data, record number and calculated result will be printed within the column. Your choices are detailed in Figure 5.

Choosing either decimals or percent results in a prompt for the number of decimal places you want to the right of the decimal point. Any number from 0 through 8 is valid. If you think a calculation might result in answers being printed in scientific notation, choose either left or right justification. If text data is formatted with a numerical format, the value of that column will be zero.

In our Inventory example, we chose left justification for text data, right for numeric and currency for the two columns of costs.

RUNNING DFCALC

Now that we have a game plan (the game gets easier the more we play), let's load DFCalc. As I mentioned earlier, you can load it from Datafile by pressing P at the main menu, and then C at the following menu, or, from Basic, by typing LOAD "DFCALC",8 <return>, and then RUN <return>.

The first display you'll see relates to configuring your printer, which I discussed earlier. The display looks like this:

```
CALCULATED REPORTS PROGRAM

USE OLD PREDEFINED FORMAT
  OR NEW FORMAT
    $ DISK DIRECTORY
    QUIT PROGRAM
TRANSFER TO:
    DATAFILE
    DFPRINT

PRESS THE APPROPRIATE KEY
```

Pressing Q terminates the program and $ displays the directory of the disk currently in the drive. Pressing O brings up the operations menu, in case a calculated report format has already been designed. Pressing N accesses the define report routines for entering all the requirements for printing the sample file Inventory. Press D. When the screen displays the following prompt, enter INVENTORY as shown, then press the return key.

```
INSERT DISK WITH THE DATAFILE
  TO BE USED FOR CALCULATED
  REPORTS.
ENTER THE NAME OF THE
  DATAFILE:
? INVENTORY
```

The program will look at the record file named Inventory and retrieve some important file-structure data that will be useful later on. The screen will then display the following sequences as you enter the data as shown:

```
DEFINE REPORT HEADER FORMAT

ENTER WIDTH OF REPORT (136 MAX)
? 70

ENTER NUMBER OF TITLE LINES IN
  THE REPORT HEADER (4 LINES MAX)
? 2

TITLE #1
? HOUSEHOLD INVENTORY

TITLE #2
? MAY 31 1985

PRESS CONT REDO START JUMP EXIT
```

After you press the return key following the second title, the mini-menu PRESS C R S J E will appear. You'll see this menu often throughout the defining sequence. It lets you change your mind as often as it appears. Here is a description of what each choice does.

C—*Continue* on to the next step. (You're satisfied with what you have done so far.) Actually, pressing any key, including return, will advance you to the next step. You'll find this useful when you're plowing back through the format.

R—*Redo* the last step. This lets you change the very last step or column. DFCalc won't allow you to go backwards more than one step or to the start of a column.

S—*Start* over. Pressing S will send you back to the screen shown above, where you started defining the format. Don't fret! You won't have to retype everything. All your entries will be preprinted on the screen for you. If you don't want to make a change, just press return to accept the value shown. Keep pressing the return key to walk back through the sequence until you get to where you left off.

J—*Jump* to the end of the definition sequence. Pressing J will send you to the operations menu, bypassing all further entry or updating of the format definition. J assumes you've already been

completely through the sequence or have loaded a previously defined format.

E—*Exit* the routine to the beginning of the program. This totally bails you out from the definitions sequence, back to where you first started the program. You might use this key if you discover you're working with the wrong record file or just want to quit.

Now, press return or any other key to continue. You'll get this display:

```
DEFINE REPORT COLUMNAR FORMAT

CALCULATE THE TOTAL LENGTH OF
  ALL FIELDS TO BE INCLUDED IN
  THE REPORT ALLOWING 2 SPACES
  BETWEEN EACH COLUMN.

ENTER THE NUMBER OF COLUMNS
  IN REPORT (16 COLUMNS MAX).
? 8

FIELD NAME      DATA FOR: INVENTORY
(LENGTH)
1 ITEM 20       4 COST 9
2 DESC 20       5 YEAR 2
3 SER# 20       6 ROOM 10
```

Before you enter 8 for the number of columns, notice the bottom of the screen. The number, name and length of the fields in the record file Inventory are displayed, so you don't have to remember! This is why you had to enter the name of the record file before starting. From now on, at least until you complete your format definition, the datafile references will stay on the screen.

IMPORTANT NOTES!

1) Don't use your cursor controls during these processes. DFCalc and *all* the Datafile programs use Input statements for entering data. Consequently, if you use your up/down cursor, the screen editor won't know where it is. To be safe, use the delete key to backspace.

2) Don't use commas, semicolons, colons or quotation marks within any input prompt, or you'll probably lose some of your data or accidentally leave the program.

Pressing the return key brings up the mini-menu (CRSJE), then pressing return again will display a prompt for the position of column 1. Enter, as you are prompted, the individual column positions: 1, 5, 16, 29, 43, 52, 56 and 61. You'll notice the column number (in reversed print) incrementing each time you press return.

The next screen asks for the heading of column 1. Enter the headings as shown on the example, pressing return each time. The screen displays the column currently being worked on, as well as the width of that column. Any header you type in that's longer than the number of characters shown will be truncated. The col-

umn headers, you recall, are: 1) #; 2) ITEM; 3) DESCRIPTION; 4) SERIAL #; 5) ORGCOST; 6) YR; 7) AGE and 8) REPCOST.

At this point we'll begin to define the individual columns as to content, justification and End of Column operations. Continuing with the next screen:

```
FORMAT INPUT ROUTINES FOR
  COLUMN 1

ENTER CONTENTS OF COLUMN 1

CHOOSE FIELD DATA
    EQUATION
    RUNNING TOTAL
    # RECORD NUMBER

    ? #
```

F will always be preprinted on the input until the fields have been defined. Remember, we want to use the record number for column 1, so press #. You'll then see this screen:

```
SET COLUMN 1 JUSTIFICATION

OPTIONS: LEFT
         RIGHT
         CURRENCY ($.00)
         PERCENT (%)
         DECIMALS TO RIGHT
? R
```

L is the default prompt. Enter R for right justification. Now, we've just completed the definition of column 1. In case you were wondering, the End of Column screen was bypassed, because the column contained only the record number, not any important data. From now on, you'll see the additional menu, and more. The next three screens reveal:

```
FORMAT INPUT ROUTINES FOR
  COLUMN 2

ENTER CONTENTS OF COLUMN 2
CHOOSE F E # Note: simplified menu
? F

WHICH FIELD # WILL BE IN
  COLUMN 2
ENTER 1 THROUGH 6
? 1


SET END OF COLUMN 2 OPERATION
OPTIONS: TOTAL SUM OF COLUMN
         AVERAGE CONTENTS OF
           COLUMN
         NO OPERATION
           TO COLUMN
? N


SET COLUMN 2 JUSTIFICATION
OPTIONS: L R C P D
? L
```

That completes column 2! Complete the same sequence for columns 3, 4, 5 and 6, according to our previous layout, changing the justification and EOC as necessary. Stop when you get to column 7.

Now we'll define the formulas, or equations, for columns 7 and 8. The following display should be on your screen:

```
FORMAT INPUT ROUTINES FOR
  COLUMN 7

ENTER CONTENTS FOR COLUMN 7

CHOOSE F E R #
? E
```

After you enter E for equation, a new screen will appear with prompts for the equation:

```
ENTER EQUATION 1 FOR COLUMN 7

RES =

OPERAND X? .
OPERATOR ? .
OPERAND Y? .
```

Note that you're prompted for equation 1. You can define up to three operations in one column. Enter the operands and operator for the equation we defined earlier for this column: 85 – F5.

Press the return key after each entry shown on the following screen. The completed equation will be shown in the conventional form, followed by the mini-menu. Press return once more to get past the prompt:

```
ENTER EQUATION 1 FOR COLUMN 7

RES =

OPERAND X? 85
OPERATOR ? -
OPERAND Y? F5

RES = 85 - F5

PRESS C R S J E
EXPAND EQUATION? Y OR N
? N
```

Since the calculation in column 7 is sufficient, we won't expand on the equation. Enter N as shown and press return. We finish defining this column by continuing through the End of Column and Justification screens. A reminder: column 7's EOC is A and justification is R.

We are now at the beginning of column 8, which will contain the equation ((F4*.05)*C7)+F4. Keep in mind that we must start at the innermost set of parentheses and work out. The entire sequence of screens for column 8 follows. (Yours should grow similarly as you keep entering data and pressing return.)

```
FORMAT INPUT ROUTINES FOR
  COLUMN 8

ENTER CONTENTS OF COLUMN 8

CHOOSE F E R #
? E


ENTER EQUATION 1 FOR COLUMN 8

RES =

OPERAND X? F4
OPERATOR ? *
OPERAND Y? .05

RES = F4*.05
```

```
PRESS C R S J E
EXPAND EQUATION? Y OR N
? Y

ENTER EQUATION 2 FOR COLUMN 8

RES = F4 * .05

OPERAND X? RES
OPERATOR ? *
OPERAND Y? C7

RES = F4 * .05 * C7

PRESS C R S J E

EXPAND EQUATION? Y OR N
? Y


ENTER EQUATION 3 FOR COLUMN 8

RES = (F4 * .05) * C7

OPERAND X? RES
OPERATOR ? +
OPERAND Y? F4

RES = ((F4 * .05) * C7) + F4

PRESS C R S J E


SET END OF COLUMN 8 OPERATION

OPTIONS: T A N
? T


SET COLUMN 8 JUSTIFICATION

OPTIONS: L R C P D
? C

PRESS C R S J E
```

When you've finished defining column 8, you've completed the entire calculated-report format.

OPERATIONS MENU

The last return you pressed presented this screen:

```
CALCULATED REPORTS
OPERATIONS MENU

CURRENT DATAFILE: INVENTORY

  SAVE REPORT FORMAT
  LOAD REPORT FORMAT
  CHANGE REPORT FORMAT
  PRINT RECORDS
  RECONFIGURE PRINTER
  $ DISK DIRECTORY
  EXIT THIS DATAFILE

PRESS THE APPROPRIATE KEY
```

Pressing $ displays the directory of the current disk in the disk drive, E exits and closes the file you've been working with and sends you back to the beginning, and C sends you to the start of the format-definition routines so you can change one or all of your entries.

You can jump back to this menu, once you've made your corrections, by pressing J at one of the many mini-menu prompts. R sends you to the printer or interface configuration menu, where you can send printer CHR$ commands, and S saves the current calculated-report format to the disk.

Warning! Any calculated-report format file (shown on the disk as CR] INV...) of the same name will be scratched and replaced with the current format. If you've made changes to a format you want to keep, and also wish to retain the original, give the revised format a new name.

Make sure you save your current format before using E or L. However, you'll be given a second chance if you happen to forget.

L loads a format file from the disk into the computer's memory. This overwrites the format currently held in memory.

THE FINAL PRODUCT

Let's assume you pressed A, for all records, at the print-options menu. The computer will print the title and heading of the report. Then it will start at the beginning of the sequential file and commence reading the records. After every record, the equations for each column will be calculated and the results printed in the current row on the report. Then the next record will be read, and so forth. When all the records have been read, the end-of-column information will be processed and printed. Finally, at the bottom of the report, the name of the record file will be printed, along with the number of records reported.■

Operands are variables (data) being operated on by operators, such as *, /, +, −, sqr and int. Operands are denoted in the List of Legal Operators (see Table 2) as X and Y. The value created by a calculation is called the resultant. To help clarify and define these terms, I'll use the following simple equations as examples:

A = B*C Z = X + Y RES = .SINX RES = X!Y RES = RES/200

In these equations, anything to the *left* of the equals sign is the *resultant* of the equation. The variables A, Z and RES are, therefore, resultants. In all the equations that you'll define using DFCalc, RES will be your resultant.

The items to the *right* of the equals sign will always consist of two operands and one operator, the middle term being the operator. B, C, X, Y, RES, 200 and the period are all operands. *, +, SIN, ! and / are all operators. The following is an expanded and more detailed description of what you can use as legal operands:

Record fields. An equation using record fields (F1, F4, F12, etc.) can take any of the following forms: RES =

F2*F3; F4!F2; F4!3; 100*F5; C3 + F3; .ABSF1; .SINF4; RES*F2; F4 + F5

The following conditions apply:

1) The number of the field defined must exist in the record file being printed.

2) The same field may be used more than once in any one column or in a number of column equations in either operand position.

3) Fields may be used in an equation with any operator except the sum through (+ +) operator.

4) The field referenced must contain numeric data. Alpha (non-numeric) characters such as A through Z, $, %, etc., in the data will produce a result of 0, an unreliable result, or an error condition when used in a calculation. If the equation cannot be resolved, "??error??" will be printed as a result or the program may crash.

Columns. An equation using columns may take the same kinds of forms as those involving fields, subject to the following condition: The column number referenced must be less than the number of the column containing the equation. If, for example, you're defining an equation for column C6, you can't use C6, C7, C15, etc., as operands.

Constants. Constants used in an equation may occupy either operand position and may have a positive or negative value, but must be within a range operable for the computer. Pi (3.14159265....) is the only predefined constant.

Other operands. The period is used as the first operand when there is only one operand being operated on. This is usually the case when a higher level operator (identified by a three-letter code) is being used. For example: .SIN 30; .INV F1; .DEG 45; .SQR 4; and so on.

RES. The resultant (RES) of a previous calculation within the current column must be used as an operand if more than one operator is used. Up to three operators or equations may be used to complete a calculation within a column. The first operator may use any two legal operands, but the second and third operators *must* have RES as one of the operands in either position.

Table 1. Rules for legal operands.

Op	Func	Description of Operation
+ +	CX + + CY	SUM the columns X through Y.
+	X + Y	ADD the values X and Y.
–	X – Y	SUBTRACT the value Y from X.
*	X*Y	MULTIPLY the values X and Y.
/	X/Y	DIVIDE the value X by Y.
↑	X↑Y	Raise the value X to the POWER indicated by the EXPONENT Y. Y may be positive, negative or fractional.
SGN	.SGNX	Changes the SIGN of X from + to – or – to +.
SQR	.SQRX	Returns the SQUARE ROOT of X.
INV	.INVX	Returns the RECIPROCAL of X (1/X).
ABS	.ABSX	Returns the ABSOLUTE VALUE of X.
INT	.INTX	Makes a fractional X into an INTEGER.
LOG	.LOGX	Returns the NATURAL (base e) LOG of X. To convert to log base 10, divide by .log10.
EXP	.EXPX	Returns the value of the mathematical constant e (2.71828. . . .) raised to the power of X.
SIN	.SINX	Returns the SINE of X, where X is measured in RADIANS.
COS	.COSX	Returns the COSINE of X.
TAN	.TANX	Returns the TANGENT of X.
SEC	.SECX	Returns the SECANT (1/COS) of X.
CSC	.CSCX	Returns the COSECANT (1/SIN) of X.
COT	.COTX	Returns the COTANGENT (1/TAN) of X.
ATN	.ATNX	Returns the ARCTANGENT (inverse tangent), or angle, in radians, whose tangent is X.
ACO	.ACOX	Returns the ARCCOSINE (inverse cosine), or angle, in radians, whose cosine is X.
ASN	.ASNX	Returns the ARCSINE (inverse sine), or angle, in radians, whose sine is X.
ASC	.ASCX	Returns the ARCSECANT (inverse secant) of X.
ACS	.ACSX	Returns the ARCCOSECANT (inverse cosecant) of X.
ACT	.ACTX	Returns the ARCCOTANGENT (inverse cotangent) of X.
RAD	.RADX	Converts the angle X from degree measure to RADIAN measure.
DEG	.DEGX	Converts the angle X from radian to DEGREE measure.

Table 2. List of legal operators.

Op	Func	Description of Operation
MMS	.MMSX	Converts the distance X from inches to MILLIMETERS.
INS	.INSX	Converts the distance X from millimeters to INCHES.
KMS	.KMSX	Converts the distance X from miles to KILOMETERS.
MLS	.MLSX	Converts the distance X from kilometers to MILES.
DFR	.DFRX	Converts the temperature X from degrees centigrade (Celsius) to degrees FAHRENHEIT.
DCG	.DCGX	Converts the temperature X from degrees Fahrenheit to degrees CENTIGRADE (Celsius).
GMS	.GMSX	Converts the weight X from ounces to GRAMS.
OZS	.OZSX	Converts the weight X from grams to OUNCES.
KGM	.KGMX	Converts the mass X from pounds to KILOGRAMS.
LBM	.LBMX	Converts the mass X from kilograms to POUNDS.
NWF	.NWFX	Converts the force X, in foot-pounds, to NEWTONS.
LBF	.LBFX	Converts the force, in newtons, to FOOT-POUNDS.
MPS	.MPSX	Converts the velocity X from feet per second to METERS PER SECOND.
FPS	.FPSX	Converts the velocity X from meters per second to FEET PER SECOND.
NSM	.NSMX	Converts the pressure X from pounds per square inch to NEWTONS PER SQUARE METER.
PSI	.PSIX	Converts the pressure X from newtons per square meter to POUNDS PER SQUARE INCH.
=	X=Y	If X equals Y then True (res=1), else False (res=0).
<	X<Y	If X is less than Y then True.
>	X>Y	If X is greater than Y then True.
<>	X<>Y	If X is not equal to Y then True.
or	XorY	If either X or Y is True (greater than 0) then True.
and	XandY	If both X and Y are True (greater than 0) then True.

When DFCalc is performing calculations, it will try to catch as many mathematical errors as possible. Some errors caused by calculations may force the computer to crash, while others may just produce inaccurate results. Syntactical errors (which cause the premature termination of the program) have, to the best of my knowledge, been accounted for.

The following produce an error condition, resulting in a value of 0 for the particular row/column cell, or an ??error?? flag.

1) Dividing by zero: X/0.
2) Taking the square root (or any even root) of a negative number: sqr(−X).
3) Taking the CSC or COT of zero: cot(0).
4) Taking the natural log of a number less than or equal to zero.
5) Raising the mathematical constant e to a power greater than 87: exp(87).
6) Using an invalid or illegal operator or operand.
7) Using an operator in an equation where an operand should be (and vice versa).
8) Using a numerical constant outside the range of the computer.
   a) Integers (whole numbers without decimal places) must be within the range of −32768 to +32767.
   b) Floating-point numbers, in scientific notation, must be within the range of +2.93873588e−39 to +1.70141183e+38.

Note: A calculation that tries to produce a result outside the legal range of the computer will result in an Overflow error and will definitely crash the program. There's no simple way to test for this, so stay away from galactic calculations. Since DFCalc uses a buffer for each column to keep track of column totals and values for calculations within a row, it's possible that your field data may inadvertently cause an Overflow error.

To help prevent what is intended to be printed as text from being interpreted as a number, you should precede suspect record-field items with an alpha character, such as # or P. This will always return a value of 0, since numbers after alpha characters are ignored.

Table 3. Causes of error.

Pick Your Ps and Qs

By Penny DeGroff

RUN It Right

C-64

Pick Your Ps and Qs is a game written for young students wishing to expand their vocabulary or for anyone who enjoys a word-game challenge.

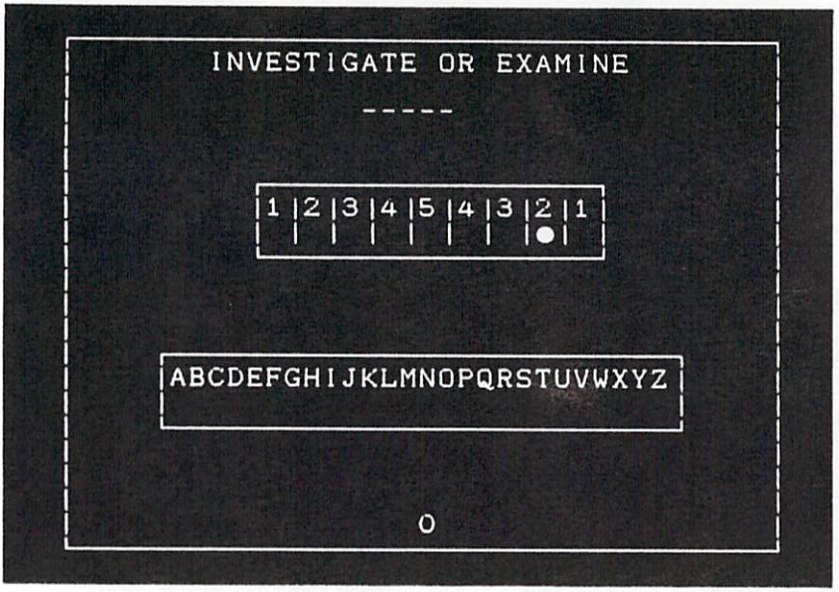
The program contains ten words and their definitions. When it's booted up, the computer chooses, at random, one word and its definition. The definition is printed near the top of the screen, and below it appears a series of dashes, one for each letter in the word. Two boxes appear in the center of the screen, one (the upper) containing the numbers 1, 2, 3, 4, 5, 4, 3, 2, 1 and the other the letters of the alphabet.

A solid ball moves from one number to another in the upper box. Using a joystick in port 2, the player presses the fire-button to stop the ball from moving. The number above the ball is the number of points the player receives if he or she picks a letter that belongs in the word.

An arrow then appears in the lower box. Using the joystick, the player moves the arrow to a letter he or she thinks is in the word and presses the fire-button. If that letter *is* in the word, it replaces one or more of the dashes, and the player's score, shown at the bottom of the screen, increases. If the player picks an incorrect letter, a buzzer sounds.

Play is continued until the entire word is revealed. If the player can guess the word without picking an incorrect letter, he or she receives a ten-point bonus, signified by a flashing border. The game is over when all ten words have been played. At that point the player's final score, along with the high score, is displayed.

You'll probably want to change the words from time to



time. To do this, list lines 710 to 800. Each line, in the form of a Data statement, contains one word, a comma and that word's definition. Move the cursor to the word to be changed and type in your replacement word, followed by a comma and the definition. Be sure to press the return key after you make the change. (Of course, if you're playing the game, you'll want to have someone else set it up for you.)

One word of caution: Don't place a comma within a definition. If you must separate the words in a definition, use a hyphen or a semicolon. Also, to keep the screen formatted properly, a definition should not contain more than 36 letters and spaces.

For a change of pace, you might set the game up to have the player find antonyms (words with opposite meanings) or synonyms (words with similar meanings).■

C-128 Screen Dump

By Trent Bills

RUN It Right

C-128 (in 40-Column mode), 1525 or compatible printer

The C-128's Basic 7.0 includes some graphics commands that make it easy for almost anyone to create his own graphic art. Unfortunately, to print out these creations, there have been only two apparent choices: to use existing Basic hi-res screen-dump programs that take 20 minutes or more; or to save the picture and switch to C-64 mode to run a 64 screen-print program. Neither alternative is especially convenient. Here is my solution to the problem.

The C-128 Screen Dump is designed to run exclusively on the C-128 in 40-Column mode. It prints hi-res or multicolor graphic screens in roughly four minutes, depending on the size of the screen, and works with a 1525 or 1525-compatible printer.

After running the program, you may call the routine at any time by pressing the F8 key. (If nothing happens, press the run-stop key and then the F8 key again.) When you call the routine, you must have an image or drawing in graphic area 1 or 2. (If you have a multicolor screen in graphic area 3 or 4, you can safely reproduce it in area 1, since the screen image will be the same as in Multicolor mode.) When you access the routine, it enables the 128's Fast (2 MHz) mode and turns off the screen. When the Print routine is finished, it turns the screen back on and returns to Slow (1 MHz) mode.

Some of you may be wondering how the program can be so short. This is because it utilizes several routines that are built into the 128's Basic interpreter. I encourage the machine language programmers among you to experiment with these Basic interpreter routines and use them in your programs. Have fun!■

Mega-Magic, March

By Michael Myers

RUN It Right

C-64; C-128 (in 40-Column mode)

File Lock—Unintentionally scratching an irreplaceable file can drive you mad. File Lock for the C-64 and C-128 (in 40-Column mode) will prevent you from scratching such files in the future by locking them on the disk and later unlocking them as you need. File Lock will lock program, sequential, user and relative files. However, it won't load locked files on the 1571 disk drive in 1571 mode.

To use File Lock, load it and then replace the disk it's on with the disk containing the programs you intend to lock.

When you run the program, the screen will display a short menu, beneath which will appear a window displaying the name of the first file on your disk. If you wish to lock it, press F5.

Using the F1 key, you move through the directory one filename at a time. Each time you reach a file you want to lock, press F5. To unlock a file, press F6 when its name appears in the window.

After you've gone through the whole directory, list it again. You'll find that the listing for each locked file includes a less-than symbol (<) to the right of the file type—PRG<, for example. When you're finished locking and unlocking files, press F7 to quit.

File Lock will also display the track (always 18) and the sector numbers (1–18) that the directory of filenames is stored on. Once you get used to which files are where on a disk, you'll be able to reach specific files quickly by using the F3 key to move to the right sector and then the F1 key to move to the file in that sector that you want.■

The Commodore In the Kitchen

By Michael Broussard

RUN It Right

C-64; C-128

Cooking has always fascinated me, because recipes are a lot like programs. After assembling and measuring ingredients, you mix them together at the right time and in the right order, and the end result is something delicious. Food recipes, however, have one major drawback: They're designed to produce a certain number of servings, and usually (at least in my house) that number is not the number needed.

If the recipe makes an extra serving, it's no big deal, unless you really hate leftovers. However, by adjusting the recipe for exactly the number of servings required, there are no leftovers, there's generally less work for the cook, and you save money as well.

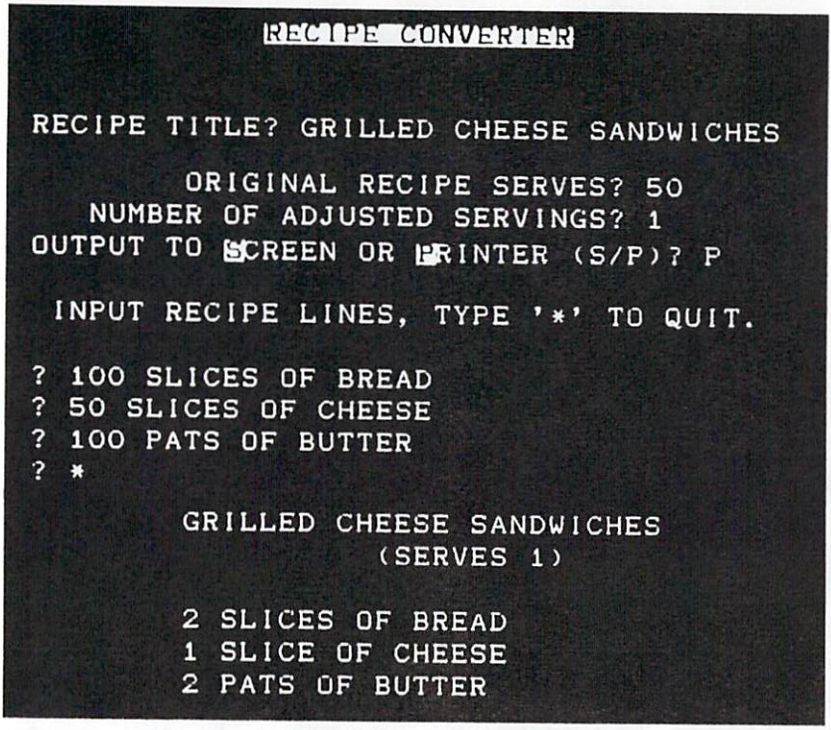
Cutting a recipe in half is pretty easy if it serves four, but adjusting a recipe for four so that it feeds six can be a little tricky. What's one-half more than ¾ of a cup?

Until now, I've tried to do these conversions in my head or with a calculator. However, unless I actually write the adjusted measurements down, I catch myself remembering to double some ingredients while forgetting to double others. I always get a sinking feeling when there isn't enough batter in the cake pans, and I don't know which ingredient I forgot to double. Worse yet, the cake then ends up sinking.

Recipe Converter (Recipe, for short) is a program I wrote that takes the guesswork out of adjusting recipes. It's written completely in Basic, so it'll run on a C-64 or a C-128 with no changes.

TYPING IN A RECIPE

When you run it, the program will first ask for the recipe title. After



you type in the name, you'll be asked how many the original recipe serves. Enter the number of servings the recipe is designed for and press the return key.

The next prompt will be for the number of adjusted servings. Type in the number of servings you want the recipe to yield and press return.

Then you'll be asked whether you want the output to go to the screen or your printer. Respond with S or P. (The program assumes your printer is device number 4, but you can change this by altering the value assigned to the variable PDEV on the first line of the listing.)

Next, the program will prompt you to enter either the recipe lines or an asterisk (*) to quit. When a question mark appears, begin typing in recipe ingredients, one per line. The program expects each line to begin with a number, followed by a measurement, followed by the ingredient

description. For example, you might type:

```
2 CUPS FLOUR
```

Recipe understands the abbreviations C and CP for cup, so you could also type any of the following:

```
2 C FLOUR
2 C. FLOUR
2 CPS FLOUR
```

All the measurements and abbreviations the program knows are listed in the Data statements in program lines 610 and 620.

As you enter each line, there will be a brief pause as Recipe examines the measurement and performs the conversion. Since the program uses the standard Basic Input statement to get recipe lines, don't use commas or colons in the lines. For example, rather than typing:

```
2 CUPS ONIONS, MINCED
```

use:

```
2 CUPS MINCED ONIONS
```

You can input fractional measurements in two ways. For example, if the recipe calls for 2¾ cups of sugar, you can enter either of these lines:

```
2.75 C SUGAR
2 3/4 C SUGAR
```

Note that in the second example there's a space between the 2 and the ¾. This space is necessary so the program won't interpret the amount as 23/4, or 5¾, cups!

Occasionally you may run into a recipe with an unusual ingredient amount, such as 1 cup plus 1 tablespoon of flour. In a case like this, just type in the amount as two different ingredient lines:

```
1 CUP FLOUR
1 TBSP FLOUR
```

THE CONVERTED RECIPE

When you've finished entering your ingredients list, type an asterisk on a line by itself to tell Recipe you're done. The converted recipe amounts will then be output to either the screen or the printer, depending on which you specified when the program started.

You'll notice that when Recipe outputs a line containing a measurement in pints or quarts, it may also include the equivalent cup measure in parentheses. For example, if you were doubling a recipe that calls for 1 cup of milk and 2 cups of flour, the output would read as follows:

```
1 PINT (2 CUPS) MILK
1 QUART (4 CUPS) FLOUR
```

The "quart of flour" appears

because Recipe has no way of knowing whether you mean liquid or dry measure when you type "cups," and though milk is indeed measured in pints, we don't usually measure flour in quarts. By providing both, Recipe lets you use the measurement that's most appropriate or convenient or customary.

Recipe will also convert ingredient amounts that aren't standard liquid or dry measurements. For example, if you're doubling 2 eggs and ¾ pound potatoes, the converted amounts would be displayed as:

```
4 EGGS
1 1/2 POUND POTATOES
```

When working with non-standard measurements, Recipe rounds to the nearest eighth. For example, if you want five servings of a recipe for six that calls for 1⅜ pounds of potatoes, you end up mathematically needing 1.1458333 pounds of spuds. Recipe takes the liberty of rounding that to 1⅛ pounds. When working with standard measures, Recipe rounds to the nearest quarter teaspoon or the nearest 5 percent, whichever is most practical.

Here are some more examples of acceptable recipe input lines:

```
1 1/2 TSP. GROUND NUTMEG
2 1/2 TBSP MELTED BUTTER
4 EGGS
1 DASH SALT
```

The following input lines, on the other hand, will cause problems, for the reasons indicated:

```
2 OR 3 CLOVES GARLIC (two measures on the same line)
DASH SALT (a number must come first)
1 TABLESP. WATER (tablesp. is not a valid abbreviation)
```

Although Recipe can't make you a good cook or guarantee that your intimate meal for two will be a success when expanded for a banquet, it will at least provide you with an accurate list of ingredient conversions. If dinner is a flop, it won't be due to your lack of math skills! ■

The Menu Machine

By E. E. Elliott

RUN It Right

C-64

Certainly, a good program needs a good menu. However, creating a menu is a tedious task at best, and is it worth spending practically as much time creating the menu as writing the rest of the program?

I wrote The Menu Machine to circumvent the problem of coding menu routines. It will take over this uninspiring part of programming for you and create impressive menus for your programs.

Each menu created with The Menu Machine appears inside a border that's sized for that menu, and the entire display is centered on the screen. You make your choices by pressing the cursor-down key until the desired selection is highlighted and then hitting the return key.

Three limitations apply when you're using The Menu Machine. First, you can include no more than eight items in each menu, although you can chain together a number of menus to effectively have more than eight choices. Second, you can't use line numbers from 20000 to 20250 elsewhere in your program unless you renumber the menu routine created by The Menu Machine. Finally, the menu routine uses the following variable names: RE, LE, TP, M, I, PL, S, K$, C, R, T, TL$, TR$, ME$(0)-ME$(8), BL$, BR$, VR$ and HZ$. If you use these names elsewhere in your program, you must be sure to reset their values after exiting from the menu routine.

You'll find it helpful to finish writing your main program before creating the menu routine, because you'll need to know where each menu choice branches to in the program. As you're creating the menu routine, MENU PRG. 1 will prompt you to enter line numbers (target lines) where execution will go after each menu selection is made.

ENTERING MENU INFORMATION

When you run MENU PRG. 1, it first asks you to enter the number of items, from one to eight,

that you want the menu to have. For a sample menu, enter 4 and press the return key.

Then you must type in the title of your menu. Enter any title up to 30 characters long.

Next, you're prompted for the text of the first menu item. For our sample, enter SELECTION 1 and press return. You're then asked for the target line number that the program should jump to when this menu item is chosen. Enter 1000. Prompts for name and target line continue through all the menu items. Enter the following information for choices two through four:

```
SELECTION 2, 2000
SELECTION 3, 3000
SELECTION 4, 4000
```

After you've completed the fourth entry, information begins flashing on the screen as the program creates your menu routine. When the cursor reappears, the program reminds you to save your menu routine before continuing. Save it to a work disk at this time.

Now clear the memory, load in MENU PRG. 2 and save it to the work disk containing the menu routine that MENU PRG. 1 created. Finally, load in MENU PRG. 3 and then save it to the same work disk.

PUTTING IT ALL TOGETHER

Now you're ready to watch The Menu Machine do its stuff. First, make sure that the work disk containing all three programs is in your disk drive. Then load in and run MENU PRG. 3 to append MENU PRG. 2 to the menu routine created by MENU PRG. 1. A prompt will appear asking Program To Append? Enter the filename of MENU PRG. 2 and press the return key. At the next prompt, specify your drive number.

When you see the cursor again, enter RUN and press the return key. At the first prompt, type in the filename of the menu routine you created with MENU PRG. 1 and press the return key. Once again, enter the device number and press return.

Finally, list the appended program and delete lines 1–6. Save the finished program to disk, and there you have it! A lean, mean, menu-driven machine.

THERE'S MORE!

As I mentioned earlier, you can create multiple menus with The Menu Machine. You do this by renumbering the menu routines as they are created. You can also link two or more menus together by making the last choice in a

menu read Next Menu or Other and using the next menu as the target line number for that choice.

When the menu routines are used in a program that requires information to be formatted on the screen, you can position your cursor anywhere by setting C equal to the column number and R equal to the row number. Then use a GOSUB 20240 to execute this feature.■

Mega-Magic, April

By Timothy M. Maloney

RUN It Right

C-64; printer

C-64 Edit Enhancer—With this utility, you can have a Quote mode toggle, a screen dump key and a screen-freeze key that will greatly enhance program editing.

THE QUOTE MODE TOGGLE

This part of the Edit Enhancer program lets you toggle Quote mode on to create reversed characters, and then off when you want to move the cursor around the screen without creating reverse characters. With a Quote mode toggle key, you can easily turn Quote mode on and off at will to help simplify editing program lines containing reverse characters.

When you're using the Quote mode toggle switch (F7), you'll know Quote mode is active by the red border on the screen; the border changes to its default color, light blue, when you deactivate Quote mode. You can select other colors by adding the following lines to the program. To change the border's default color (Quote mode off), add 225 POKE BB + 49,{color code #}. To change the border's red color (Quote mode on), add 226 POKE BB + 57,{color code #}.

To demonstrate the Quote mode toggle's usefulness, load and run 64 Edit Enhancer. Then type in the following line: 10 PRINT"CDTHIS IS A TESTR" and press return. Now try replacing the C, D and R characters with reverse characters. A pain, isn't it? Next, position the cursor over the C, and press F7 to turn Quote mode on (notice the border-color change). Then press the shifted-clear key, followed by the cursor-down key. The reverse characters of the keys should replace the C and D.

Press the F7 key again to turn Quote mode off, and use the cursor-right key to move the cursor to the final R. Press F7 again, followed by the cursor-right key. Finally, press return to enter the

line. Need I say more about the advantages of the Quote mode toggle?

THE SCREEN DUMP

The F8 key activates the screen dump and can be used with most printers—graphic or nongraphic. You can use it any time you want to dump a screen to the printer.

When you run the program, it will ask if your printer is ready. Turn on your printer and answer Y. (An N answer implies that you don't have a printer and will disable the screen dump feature.) The program will then ask if your printer supports graphics characters. Answer Y if your printer (and interface) can emulate the Commodore graphics characters; otherwise enter N. If your printer cannot produce graphics characters, a space is substituted for every graphics character the program encounters.

THE SCREEN-FREEZE

If you want to print a frozen screen, first pause the screen momentarily with the shift key (or indefinitely with the shift-lock key). Then press F8 to print the screen. When the shift key is released, the program will continue where it left off.

The machine language program is poked in starting at location 52900, but, if necessary, you can poke it into any available 347-byte memory space by changing the value of BB in line 30 to the new starting address. ■

Please send me back issues of ReRUN

____ Fall Edition *
____ Winter Edition *
____ January/February 1986
____ March/April 1986
____ May/June 1986
____ July/August 1986
____ Productivity Pak II
____ September/October 1986
____ November/December 1986
____ January/February 1987

____ Cassette version(s) at $11.47
____ Disk version(s) at $21.47

Prices include postage and handling. For foreign air mail, please add U.S. $1.50 per item and $25 per subscription. Prepayment only.

Includes programs for C-64 and C-128 (in both 64 and 128 modes).

☐ Payment Enclosed ☐ MC ☐ VISA ☐ AE

**Available on cassette only.*

Card # Exp. Date

Name

Address

City State Zip

Signature

ReRUN • 80 Elm Street • Peterborough, NH • 03458

BEAT THE RUSH!

Please send me:

☐ **1 year (6 issues) for $89.97**
☐ **May/June 1987 ReRUN disk for $21.47.***

Available in June 1987.

Price includes postage and handling. For foreign air mail, please add U.S. $1.50 per item and $25 per subscription. Prepayment only.

☐ Payment Enclosed ☐ MC ☐ VISA ☐ AE

Card # Exp. Date

Name

Address

City State Zip

Signature

ReRUN • 80 Elm Street • Peterborough, NH • 03458

9 *RUN* Programs Included on this Disk:

Database Management ▸ Education ▸
Recipe Converter ▸ Easy Applications ▸
Menu Utility ▸ Disk File Lock

From the March *RUN*:

- DFPrint
- Pick Your Ps and Qs
- C-128 Screen Dump
- Mega-Magic (File Lock)
- The Commodore in the Kitchen

From the April *RUN*:

- DFCalc
- The Menu Machine
- Mega-Magic (64 Edit Enhancer)

PLUS Bonus Program:

- Datafile 3.6

If any manufacturing defect becomes apparent, the defective disk will be replaced free of charge if returned by prepaid mail within 30 days of purchase. Send it, with a letter specifying the defect, to:

ReRUN • 80 Elm Street • Peterborough, NH 03458

Replacements will not be made if the disk has been altered, repaired or misused through negligence, or if it shows signs of excessive wear or is damaged by equipment.

The programs in ReRUN are taken directly from listings prepared to accompany articles in *RUN* magazine. They will not run under all system configurations. Use the RUN It Right information included with each article as your guide.

The entire contents are copyrighted 1987 by CW Communications/Peterborough. Unauthorized duplication is a violation of applicable laws.

©Copyright 1987 CW Communications/Peterborough